

## ADDER-SAVING IMPLEMENTATION OF DIGITAL INTERPOLATION/DECIMATION FIR FILTER

### **BACKGROUND OF THE INVENTION**

### **FIELD OF THE INVENTION**

This invention relates generally to digital filters and, more particularly, to a novel architecture for a finite impulse response (FIR) filter.

### **BRIEF DESCRIPTION OF THE PRIOR ART**

Digital filters are well known in the prior art. Such filters receive sampled digital signals and transmit a sampled waveform therethrough. The waveform transmitted by the digital filter is determined by coefficients operating on portions of the transmitted digital signal. A typical prior art digital filter has a plurality of serially connected delay components with an output of each delay component transmitted to a coefficient addition component, the coefficient addition component adding the output from the delay component applied thereto by a weighting factor derived from a transform function. The outputs of the coefficient addition components are applied to the input of a succeeding delay element and eventually provide the filter output signal. Accordingly, an input signal, after an appropriate delay, is filtered according to the coefficient addition components with the resulting signal being applied to the digital filter output.

A typical prior art FIR filter is shown in FIGURE 1 which shows the register-level implementation diagram for a seven-tap FIR filter which operates in accordance with the equation:

$$H(z) = b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5} + b_6z^{-6} + b_7z^{-7}$$

where  $H(z) = Y(z)/X(z)$  is the transfer function of the system. For a linear-phase response, the coefficients must be symmetric with respect to the center tap with  $z^{-1}$  representing a register unit (such as a D flip-flop) to store the result of the previous calculation. The input data  $x(n)$  can be interpolated by, for example, 2 before filtering in the present example, though this number is arbitrary. Interpolation by  $M$  means that  $M - 1$  zeros are inserted between adjacent input samples. On the other hand, decimation by  $M$  means that every  $M - 1$  input samples are dropped. Digital filters which perform these functions are known as interpolation/decimation filters. The interpolation/decimation filters are commonly used in modern digital communication and audio systems. Assuming that the input data rate is  $\text{clk}$ , then the clock rate for the FIR filter in FIGURE 1 has to be twice  $\text{clk}$  due to the interpolation at the front or at the filter input.

The equation for any FIR filter  $H(z)$  can be decomposed into  $Z^{-1}H_0(z) + H_1(z)$  where:

$$H_0(z) = b_1 + b_3z^{-2} + b_5z^{-4} + b_7z^{-6}$$

and  $H_1(z) = b_2z^{-2} + b_4z^{-4} + b_6z^{-6}.$

Therefore, the system of FIGURE 1 can be upgraded into a system as shown in FIGURE 2 which includes two serially connected sets of delay elements (e.g. D flip-flops), each delay element except the first is preceded by an adder which adds in one of the coefficients. One set of delay elements includes a first delay element which receives the input  $x(n)$  and multiplies the input by the coefficient  $b_7$  at a first delay, the output of which is added to the input multiplied by the coefficient  $b_5$  and delayed with the output of that delay being added to the input multiplied by the coefficient  $b_3$  and being delayed, the output thereof being added to the input multiplied by the coefficient  $b_1$  and being delayed with the output of the final delay being multiplexed with the output of the other set of delay elements to provide the output  $Y_n$  of the filter. The second set of

delay elements also receives the input  $x(n)$  and multiplies the input by the coefficient  $b_6$  at a first delay, the output of which is added to the input multiplied by the coefficient  $b_4$  and delayed with the output of the delay being added to the input multiplied by the coefficient  $b_2$  and delayed with the output of that delay being added to zero and delayed to the output for multiplexing as discussed above. It can be seen that the input  $x(n)$  is applied to both sets of delay elements with the outputs of the two sets of delay elements being multiplexed to provide output  $y(n)$ .

The advantage of the system of FIGURE 2 over that of FIGURE 1 is that the clock rate of the filter is the same as the input data rate  $\text{clk}$ , which is half of the clock rate of the system of FIGURE 1. However, the system of FIGURE 2 has two basic disadvantages, these being (1) that the amount of hardware, including adders and registers, is no less than that in the FIGURE 1 system and (2) that the multiplexer connecting the output of each set of delay elements may cause glitches in the final filter output,  $y(n)$ . More specifically, there are the same number of adders and one more register used in the FIGURE 2 system wherein the adder for adding zero is not actually needed, but the associated register is necessary in order to maintain the correct output. For even-order FIR filters, the number of adders and registers is the same for the FIGURE 1 and FIGURE 2 systems. With reference to the multiplexer, in digital design, it is desirable to obtain the output from a register rather than from combinational circuits. Placing the multiplexer within the register can resolve the glitch issue, however it will introduce more gates into the circuits, which is not desirable.

## SUMMARY OF THE INVENTION

Because the output of the Mth order FIR filter depends upon its previous M input data, the number of registers is usually no less than M. However, a number of adders can be saved (eliminated) through the resource of reuse, which is an important feature of the present invention. Since an N-bit ripple adder takes about the same number of gates as does an N-bit register, reducing the number of adders also can reduce quite significantly the silicon area of the semiconductor chip required for the filter.

The second set of delay elements of FIGURE 2 is identical in structure to the first set of delay elements. Therefore, in accordance with the present invention, the adders in the first set of delay elements are reused or, in other words, these adders are used in conjunction with both the first and second set of delay elements. To this end, the filter is operated at twice the clock rate of clk. In the first half of the clk period, the output of the first set of delay elements is calculated and in the second half of the clk period, the output of the second set of delay elements is calculated. The output  $y(n)$  is the output of the last register in the first set of delay elements with no output multiplexer required. In summary, the adders of the first set of delay elements are reused by running the filter at twice the clock rate and computing  $H_0(z)$  and  $H_1(z)$  on a time-sharing basis.

The system can be further optimized by omitting two redundant registers as shown in FIGURE 4. The first delay element in each set of delay elements  $Q_1$  and  $Q_2$  can be shared, thereby removing one of the delay elements. Also, the last delay element in each set of delay elements  $Q_7$  and  $Q_8$  can be shared, thereby removing another delay element. Since the delay element  $Q_7$  stores the quantity associated with the coefficient  $b_7$  in the first clock phase, in order to share  $Q_7$  in the second clock phase, the corresponding coefficient needs to be adjusted by

subtracting  $b_7$  from  $b_6$  to obtain the correct result. Multiplication of  $x$  by  $b_i$  is actually performed by shifts and addition of  $x$  in standard manner.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

FIGURE 1 is a diagram of a prior art FIR filter;

FIGURE 2 is a diagram of an upgraded FIR filter in accordance with the prior art;

FIGURE 3 is a diagram of an FIR filter in accordance with a first embodiment of the present invention;

FIGURE 4a is a diagram of an FIR filter in accordance with a second embodiment of the present invention;

FIGURE 4b is a block diagram of an FIR filter in accordance with a second embodiment of the present invention; and

FIGURE 5 is a timing diagram for use in conjunction with the FIR filter of FIGURE 4b.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference first to FIGURE 3, there is shown a first embodiment in accordance with the present invention. In accordance with this embodiment, adders are saved through reuse. As stated above, since an N-bit ripple adder takes about the same number of gates as does an N-bit register, reducing the number of adders can also reduce the silicon area of the semiconductor chip required for the components quite significantly. Since the second set of delay elements of FIGURE 2 is identical in structure to the first set of delay elements, the reuse of adders in the first set of delay elements can be effectuated. To this end, the filter is operated at twice the clock rate of clk. In the first half of the clk period, the output of the first set of delay elements  $Q_7$ ,  $Q_5$ ,  $Q_3$  and  $Q_1$  is calculated and in the second half of the clk period, the output of the second set of delay elements  $Q_8$ ,  $Q_6$ ,  $Q_4$  and  $Q_2$  is calculated as shown in FIGURE 3. The output  $y(n)$  is the output of the last register in the first set of delay elements with no output multiplexer required. The adders of the first set of delay elements are reused by running the filter at twice the clock rate and computing  $H_0(z)$  and  $H_1(z)$  on a time-sharing basis using only the adders shown as coupled to the delay elements  $Q_7$ ,  $Q_5$ ,  $Q_3$  and  $Q_1$  on a multiplexed basis.

The system of FIGURE 3 can be further optimized to omit two redundant registers. In this regard, delay elements  $Q_1$  and  $Q_2$  can be shared, thereby removing delay element  $Q_2$ . Also delay element  $Q_8$  can be shared with delay element  $Q_7$ , thereby removing delay element  $Q_7$ . Since delay element  $Q_7$  stores  $b_7x$  in the first clock phase, it has to be adjusted by subtracting  $b_7x$  in order to obtain the correct value in the second clock phase. This system is shown in FIGURE 4. Multiplication of  $x$  by  $b_i$  is actually performed by shifts and addition of  $x$ .

The example as set forth in FIGURE 4 is an interpolation filter with interpolation factor of two, it being understood that the invention herein can be applied to a decimation filter as well.

A decimation filter is similar to an interpolation filter in its implementation except that, with reference to FIGURE 2, the multiplex output turns into the input multiplexer in the front for the decimation filter. This invention can also be applied to filters with interpolation/decimation factor other than two, the number two being used herein only by way of example.

The silicon area saved by the present invention for an Lth order filter with decimation factor M can be estimated. The major hardware for the filter is assumed to be the adders and the registers. It is also assumed that an N-bit adder has the same gate count as an N-bit register. The total number of registers required is NL, and the adders equivalently occupy NL register areas. Therefore, the total number of areas required for the system of FIGURE 1 is 2 NL. In contrast, there are only N/M adders in the system of FIGURE 4b, therefore the total number of areas required is  $(1 + 1/M)NL$  for the system of FIGURE 4b. For the foregoing example, with  $M = 2$ , the required chip area is reduced by 25 percent by this estimation. Greater chip area reduction can be achieved with a higher interpolation/decimation factor M. A 7th order filter with  $M = 2$  can be synthesized with a 24 percent reduction in gate count. There are slightly more gates required than the estimation, mainly due to the small number of multiplexers required for the coefficients as shown in FIGURE 4b.

With reference to FIGURE 4b, the input x is applied to each of a first multiplier where it is multiplied by the coefficient  $b_7$ , a second multiplier where input x is multiplied by result of coefficients  $b_7 - b_6$ , a third multiplier where input x is multiplied by coefficient  $b_5$ , a fourth multiplier where input x is multiplied by coefficient  $b_4$ , a fifth multiplier where input x is multiplied by coefficient  $b_3$ , a sixth multiplier where input x is multiplied by a coefficient  $b_2$  and a seventh multiplier where input x is multiplied by a coefficient  $b_1$ . The  $b_7$  product is delayed by D flip-flop  $Q_7$ , which is clocked at a clock rate of  $clk/2$  and supplied to a first adder. The  $b_6 - b_7$



product and the  $b_5$  product are multiplexed by the clock  $clk$ , the output of which is added to the output of flip flop  $Q_7$  in the first adder to provide an output  $S_1$ . This output is delayed in D flip flops  $Q_6$  and  $Q_5$  which are also clocked at the clock rate  $clk/2$  with the output applied to a second adder. The  $b_4$  product and  $b_3$  product are multiplexed by the  $clk$ , the output of which is applied to the second adder. The sum of the two signal applied to the second adder is the output  $S_2$  which is applied to D flip-flops  $Q_4$  and  $Q_3$  which are also clocked at a clock rate of  $clk/2$  with the output of these flip-flops being applied to a third adder. The  $b_2$  and  $b_1$  products are multiplexed by the  $clk$ , the output of which is also applied to the third adder and added to provide the output  $S_3$ . This output is delayed by D flip-flop  $Q_2$  which is clocked at the clock rate  $clk/2$  to provide the output  $y$ . The timing diagram for the circuit of FIGURE 4b is shown in FIGURE 5 where all registers are clocked by  $clk/2$  while all multiplexers are controlled by  $clk$  which operates at one half the speed of  $clk/2$ . The following equations are true for the embodiment of FIGURE 5:

$$\begin{aligned} \text{At phase 0: } & S1(i) = b_7x(i) + (b_6 - b_7) x(i) = b_6 x(i) \\ & S2(i) = S1(i-2) + b_4x(i) = b_6x(i-2) + b_4x(i) \\ & S3(i) = S2(i-2) + b_2x(i) = b_6x(i-4) + b_4x(i-2) + b_2x(i) \\ \\ \text{At phase 1: } & S1(i) = b_7x(i-2) + b_5x(i) \\ & S2(i) = S1(i-2) + b_3x(i) = b_7x(i-4) + b_5x(i-2) + b_3x(i) \\ & S3(i) = S2(i-2) + b_1x(i) = b_7x(i-6) + b_5x(i-4) + b_3x(i-2) + b_1x(i) \end{aligned}$$

where  $i$  is the time index and  $i-n$  means  $n$  time clocks later. In phase 0, the result of the upper half branch in FIGURE 4b is obtained and in phase 1 the result of the lower half branch in FIGURE 4b is obtained.

It can be seen that the function of the circuit of FIGURE 2 is provided in FIGURES 4 and 5 without four of the adders, two of the flip-flops and without the multiplexer. Accordingly, a significant amount of chip area is saved for other uses.

Though the invention has been described with reference to a specific preferred embodiment thereof, many variations and modifications will immediately become apparent to those skilled in the art. It is therefore the intention that the appended claims be interpreted as broadly as possible in view of the prior art to include all such variations and modifications.